

A pragmatic approach to distributed nonlinear model predictive control: Application to a hydrostatic drivetrain

Abhishek Dutta^{*,†}, Clara Ionescu and Robain De Keyser

Electrical Energy, Systems and Automation, Ghent University, Sint-Pietersnieuwstraat 41 Block B2, 9000 Gent, Belgium

SUMMARY

The global control of large-scale production machines composed of interacting subsystems is a challenging problem due to the intrinsic presence of high coupling, constraints, nonlinearity, and communication limitations. In this work, a pragmatic approach to distributed nonlinear model predictive control (DNMPC) is presented with guaranteed decrease in cost. Furthermore, in order to tackle time-varying process dynamics, a learning algorithm is developed, thereby improving the performance of the global control. The proposed control framework is experimentally validated on a hydrostatic drivetrain, which exhibits nonlinear dynamics, strongly interacting subsystems. The experimental results indicate that good tracking performance and disturbance rejection can be obtained by the proposed DNMPC. Copyright © 2014 John Wiley & Sons, Ltd.

Received 23 August 2013; Revised 15 March 2014; Accepted 29 July 2014

KEY WORDS: distributed control; nonlinear control; predictive control; hydrostatic drive

1. INTRODUCTION

The ever increasing complexity of large-scale systems found nowadays in process industry, manufacturing systems, and traffic networks urged the control community to revise old concepts of distributed control and develop novel, pragmatic approaches. An excellent review of the current techniques used in practice is given in [1]. The challenge for control is that these large-scale systems are composed of many interacting subsystems. They can be difficult to control with a linear centralized control structure due to nonlinearity, computational complexity, and limitations on communication [2, 3].

Many industrial systems can be described by a hierarchical structure where an algorithm at higher level coordinates the actions of local regulators placed at a lower level. However, often, the high level algorithm becomes so complex that it becomes hard to justify its advantages over a centralized controller [1]. For all these reasons, in the last decade, many distributed control structures have been developed and the nonlinear model predictive control (MPC) approach was recognized as one of the most suitable candidates [4, 5]. This is not surprising, because MPC has a great potential to play a crucial role in distributed control due to its intrinsic forecasting properties, which can be exchanged (in part) to neighboring MPC units [1, 6–9].

In this paper, we introduce a pragmatic approach to distributed MPC by taking into account the interaction of the subsystem with minimum amount of exchanged information with a guaranteed improvement in cost. We make use of the in-house developed nonlinear extended prediction self-adaptive control (NEPSAC) algorithm as a basis for stemming the proposed approach [10]. The novelty of our approach is the ease of implementation preserving the guarantee of improvement in cost, pragmatism, and ability to tackle constraints without significant computational

^{*}Correspondence to: Abhishek Dutta, Electrical Energy, Systems and Automation, Ghent University, Sint-Pietersnieuwstraat 41 Block B2, 9000 Gent, Belgium.

[†]E-mail: ad711@sref.ucam.org (Wolfson College Cambridge)

complexity. In order to test these claims, a hydrostat drivetrain system is used, consisting of two highly nonlinear, time-varying dynamic, interacting subsystems. This is a representative global mechatronic system widely used in industry, for example, in mobile vehicles such as ground moving machines, agricultural machines, forest machines, and industrial and mining lifters. The use of hydrostatic transmission as the vehicle drives is primarily motivated by its large range of continuously variable speed, high maneuverability, and a possibility to increase the overall efficiency [11].

This motivates us to develop a distributed nonlinear model predictive control (DNMPC) framework, which can guarantee an improvement in cost with every cycle for the distributed nonlinear control of the hydrostat, and to the best of our knowledge, this opens up a new way of viewing and controlling such global production machines. The paper elaborates on this practical research and is organized as follows. Section 2 introduces the DNMPC technique with distributed recursive least squares (DRLS). Section 3 presents a physical model and open-loop characteristics of the hydrostat. The presented technique is then experimentally validated on the hydrostat in Section 4 by comparing it with a PID and is followed by the principle conclusions in Section 5.

2. DISTRIBUTED NONLINEAR MPC (DNMPC)

Within the distributed control framework, a limited amount of information is transmitted between the local regulators, such that each of them has some knowledge about the behavior of its interacting neighbors. For simplicity, a schematic overview is given in Figure 1, depicting the concept of distributed MPC with interacting subsystems. The information exchange can be performed either

- Non-iterative, that is, information is transmitted/received only once in each sampling period; or
- Iterative, that is, information is transmitted/received many times to reach global consensus within each sampling period.

If the information is part of the cost function, then this can be either

- Independent, that is, each regulator minimizes a local performance index; or
- Cooperating, that is, all local regulators minimize a global cost function.

Agent-based distributed MPC scheme is another approach that is based on cooperative game theory and has been reported for constrained linear systems in [12]. In this paper, we will explore the cooperative approach due to its established stability properties [13] along with the non-iterative scheme to be able to perform fast computation and give an extension for nonlinear systems with a guarantee on monotonic decrease in global cost.

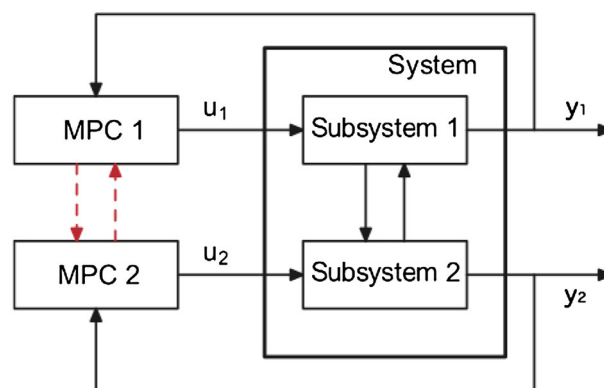


Figure 1. Distributed MPC of two interacting subsystems with information exchange.

2.1. Proposed DNMPC algorithm

In this section, we elaborate on the DNMPC controller design. To facilitate the exposition, we assume that the plant comprises only two subsystems. For a process with two inputs and two outputs, the structure of the generic process model becomes [14]

$$y_1(t) = \hat{y}_1(t) + n_1(t) \quad \text{and} \quad y_2(t) = \hat{y}_2(t) + n_2(t) \quad (1)$$

with $y_i(t)$, $\hat{y}_i(t)$, $n_i(t)$ as the i^{th} process output, model output, and disturbance, respectively, where

$$\begin{aligned} \hat{y}_1(t) &= f_1[\hat{y}_1(t-1), \hat{y}_1(t-2) \dots u_1(t-1), u_1(t-2) \dots u_2(t-1), u_2(t-2)] \\ \hat{y}_2(t) &= f_2[\hat{y}_2(t-1), \hat{y}_2(t-2) \dots u_1(t-1), u_1(t-2) \dots u_2(t-1), u_2(t-2)] \end{aligned} \quad (2)$$

in the case where the system is available in the input–output formulation. In the case of a state-space realization,

$$x_1(t+1) = f_1^x(x_1(t), x_2(t), u_1(t), u_2(t)), \quad x_2(t+1) = f_2^x(x_1(t), x_2(t), u_1(t), u_2(t)) \quad (3)$$

$$y_1(t) = g_1(x_1(t), x_2(t)) + n_1(t), \quad y_2(t) = g_2(x_1(t), x_2(t)) + n_2(t) \quad (4)$$

The disturbances are modeled by colored noise processes

$$n_1(t) = (C_1(q^{-1})/D_1(q^{-1})) \cdot e_1(t) \quad \text{and} \quad n_2(t) = (C_2(q^{-1})/D_2(q^{-1})) \cdot e_2(t) \quad (5)$$

where $e_1(t)$, $e_2(t)$ are zero mean white noise sequences. In the linear case, the future response can then be expressed as

$$y_1(t+k|t) = y_{1base}(t+k|t) + y_{1opt}(t+k|t) \quad \text{and} \quad y_2(t+k|t) = y_{2base}(t+k|t) + y_{2opt}(t+k|t) \quad (6)$$

where the predictions are made at time t over the prediction horizon $k = \overline{N_1, N_2}$. In vector notation,

$$Y_1 = \bar{Y}_1 + G_{11} \cdot U_1 + G_{12} \cdot U_2 \quad \text{and} \quad Y_2 = \bar{Y}_2 + G_{21} \cdot U_1 + G_{22} \cdot U_2 \quad (7)$$

where \bar{Y}_1, \bar{Y}_2 are the base responses computed as the cumulative effect of both the past control inputs/outputs (states), the a priori defined future control actions U_{1base}, U_{2base} , and the predicted disturbances. The rest of the optimizing terms are the discrete time convolution of the predicted optimal inputs U_1, U_2 (defined as the increments to U_{1base}, U_{2base}) with the corresponding impulse response coefficients, that is, the respective G_{ij} matrices defined from input j to output i , where $i, j \in 1, 2$. Thus, $U_1 = [\delta u_1(t|t) \dots \delta u_1(t + N_{u1} - 1|t)]^T$, where N_{u1} is the control horizon and $\delta u_1(\cdot)$ is the control increments from the base input.

Next, consider the following global cooperating cost function

$$V = (V_1(U_1, U_2) + V_2(U_1, U_2)), \quad \text{where,} \quad V_i = (R_i - Y_i)^T \cdot (R_i - Y_i) + U_i^T \cdot \Lambda_i \cdot U_i \quad (8)$$

where R_i, Λ_i are the respective reference trajectories and control penalty matrices for $i = 1, 2$. It follows that the optimization problem for MPC-1 is

$$U_1^* = \text{Min}_{U_1} V, \quad \text{subject to} \quad U_1 \in U_1^c \quad \text{and} \quad U_2 = U_{2base}^{*-1} \quad (9)$$

where U_{2base}^{*-1} is the optimal input trajectory communicated by MPC-2 delayed by one sample, and U_1^c is the polytopic constraint set for MPC-1, which arises from the input and output constraints of the model. An explicit solution can be obtained in the unconstrained case

$$\begin{aligned} U_1^* &= (G_{11}^T \cdot G_{11} + G_{21}^T \cdot G_{21} + \Lambda_1 \cdot I)^{-1} \cdot (G_{11}^T \cdot (R_1 - \bar{Y}_1) \\ &\quad + G_{21}^T \cdot (R_2 - \bar{Y}_2)) \end{aligned} \quad (10)$$

Similarly, an explicit solution can be derived for U_2^* . Note that in many cases, a control horizon of $N_u = 1$ sample suffices and then we can still use the explicit solution followed by clipping. A short control horizon is widely used in the industry, at least in the case of stable plants when only input constraints are active. When the underlying process model is nonlinear, for controller-1, the superposition of (7) is still valid only if the term $y_{1opt}(t+k|t)$ is small enough compared with

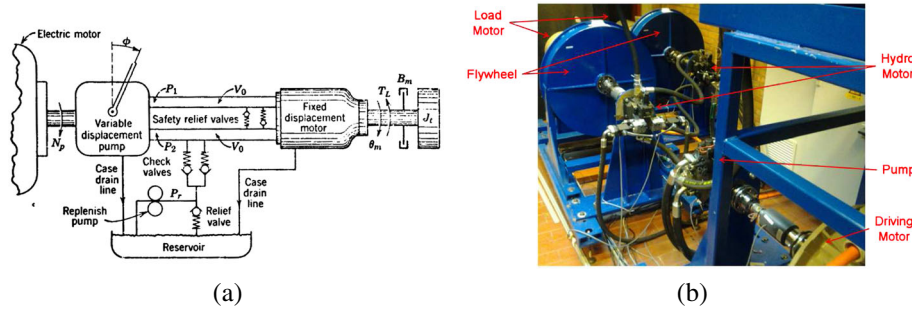


Figure 2. (a) A schematic of the pump-controlled motor, (b) hydrostat benchmark consisting of two hydromotors driven by a pump.

$y_{1base}(t+k|t)$. This is true when $\delta u_1(t+k|t)$ is small, which is the case if $u_{1base}(t+k|t)$ is close to the optimal $u_1^*(t+k|t)$. To address this issue, the idea is to recursively compute $\delta u_1(t+k|t)$ using (10), within the same sampling period, until $\delta u_1(t+k|t)$ converges to 0. Inside the recursion, $u_{1base}(t+k|t)$ is updated each time to $u_{1base}(t+k|t) + \delta u_1(t+k|t)$, that is, the extension of extended horizon self-adaptive control (EPSAC) algorithm for nonlinear systems [10, 14]. Notice that linearization of the process is not necessary in this case, which is a significant advantage over other MPC strategies. The procedure is similar for controller 2, and both controllers are further denoted in the remainder of this paper as NMPC-1 and NMPC-2.

To summarize, the non-iterative sequential DN MPC algorithm for the 2×2 process consists of the following steps:

1. Initialize the control sequences U_{1base}, U_{2base} and compute the respective disturbance signals n_i , with $i = \overline{1, 2}$;
2. NMPC-1: Compute $U_1^* = \arg \min_{U_1} V$ subject to $U_1 \in U_1^c$ and $U_2 = U_{2base}^{-1}$. Update $U_{1base} = U_{1base} + U_1^*$ and communicate to NMPC-2 together with $n_1(t)$.
3. NMPC-2: Compute $U_2^* = \arg \min_{U_2} V$ subject to $U_2 \in U_2^c$ and $U_1 = U_{1base}^*$. Update $U_{2base} = U_{2base} + U_2^*$ and communicate to NMPC-1 together with $n_2(t)$.
4. NMPC-1 and NMPC-2 apply the first inputs of U_{1base}, U_{2base} to the real process.
5. Go to step 1 at the next sampling period.

Algorithm 1: Non-iterative sequential DN MPC

There are two main advantages of the proposed strategy. First, if a temporary failure in communication occurs, NMPC-1 can compute both U_1^* and estimate of U_2^* (and similarly for NMPC-2), whereas only the disturbance estimates are exchanged until communication is re-established. Second, for large-scale systems with for instance 50 decision variables, the resulting control horizon may be 50, which requires large matrix inversion and complex quadratic programming optimization. However, the DN MPC algorithm simplifies it to 50 subsystems, each optimizing over the control horizon $N_u = 1$. In this case, the explicit solution can be used, which reduces the computational burden to a scalar division.

A centralized NMPC (CNMPC), that is, multivariable NEPSAC would minimize the same cost function of (8) but now with respect to the multivariable vector $[U_1, U_2]^T$ at once subject to the constraints $[U_1 \times U_2]$. This, even in the limiting case, that is, when both the inputs have individual horizons of 1, adds up to a net control horizon of 2 and hence can only be solved by quadratic programming, which has an exponential cost. Comparatively, in this case, the DN MPC with control horizons 1 will have a polynomial time complexity.

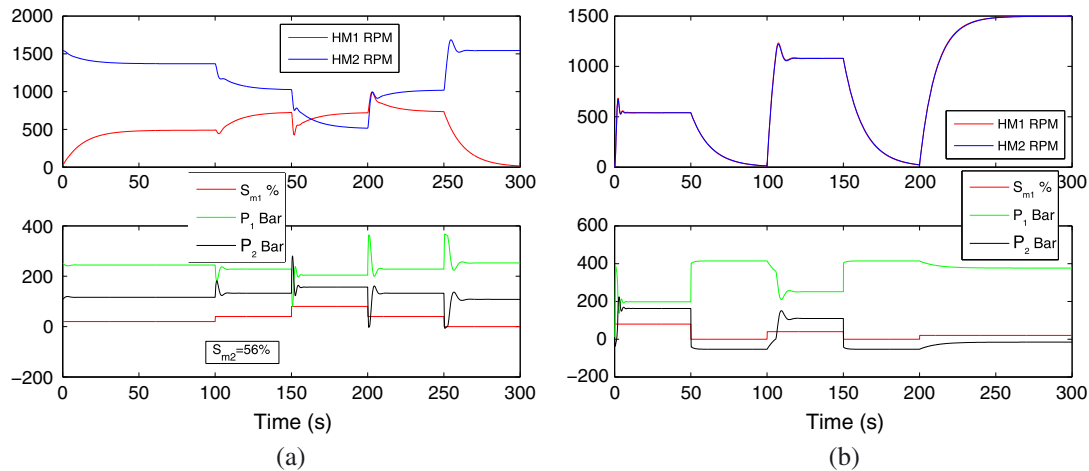


Figure 3. (a) Open-loop test on the hydrostat model demonstrating the coupled dynamics and (b) the presence of significant nonlinearity.

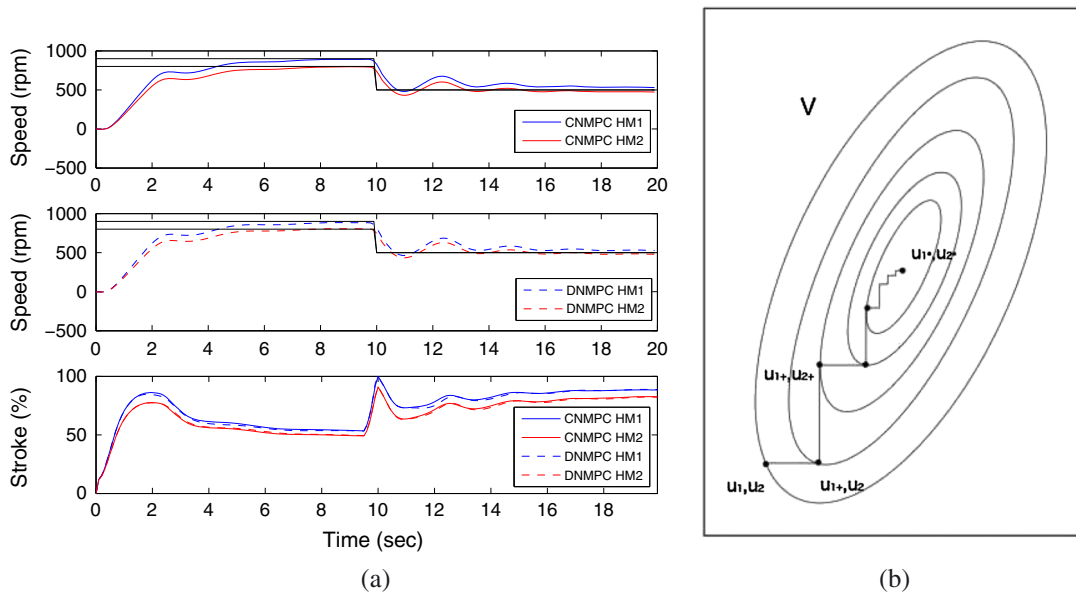


Figure 4. (a) A comparison of centralized and distributed NMPC for tracking; (b) the sequential distributed NMPC algorithm approaching centralized NMPC locally optimal solution.

Assumption 1

The cost function of (8) is smooth and twice differentiable.

Theorem 1

The presented sequential non-iterative DNMP algorithm 1 achieves a guaranteed decrease in global cost with every iteration.

Proof

A nonlinear system usually leads to a non-convex cost function, which is convex around a neighbourhood where the local minimum exists. In such a neighbourhood, let us say u_1, u_2 are the current iterates. The presented DNMP algorithm then freezes one direction and takes a gradient descent step in the other direction to u_1^+, u_2 as shown in Figure 4(b). Next, the converse happens and a gradient descent step is taken in the other coordinate to u_1^+, u_2^+ . The monotonic decrease in the cost function due to NEPSAC can be ensured by a suitable choice of control penalty Λ , which is akin to

Levenberg–Marquardt algorithm for each controller and together is called coordinate descent. The decrease can only be 0 on both directions if a stationary point is attained which is the local optimal for the global cost. (A Taylor's series expansion around initial point $V(U_{1base}, U_{2base})$ leads to $V(U_{1base} + U_1, U_{2base})$, which is minimized to obtain (10) and so on.) \square

Lemma 1

If the steps 2,3 of algorithm 1 are iterated over and over, convergence is guaranteed to the solution of the CNMPC.

Proof

The satisfaction of Assumption 1 (which is assumed) guarantees that the algorithm does not get stuck in corners of the level sets of the global cost function, and hence, the coordinate descent would stop decrementing only at a locally optimal solution (u_1^*, u_2^*) with cost $V^*(U_{1base}^*, U_{2base}^*)$, same as the one obtained when the optimization is performed in a multivariable fashion by CNMPC. This is shown in Figure 4(b). \square

2.2. Distributed learning

The production machines have intrinsic time-varying dynamics (e.g., oil temperature, density, and leakage). They are also intensively operated under varying environmental conditions (e.g., process properties, toxic gas, and seasonal variations). Consequently, these factors imply the necessity of an adaptation mechanism for updating the model parameters in a distributed sense. In this section, we propose a simple yet effective learning method.

The model equation for the i^{th} subsystem can be written as

$$y_i(t) = \phi_i^T(t) \cdot \theta_i(t) + \phi_{i-}^T(t) \cdot \theta_{i-}(t) + v_i(t) \quad (11)$$

where the nonlinear system is assumed to be linear in the parameter θ . The vector ϕ contains all the past inputs and measurements (i.e., state), v denotes the error and subscript $i-$ denotes all other parameters except the i^{th} . We employ the classic recursive least squares (RLS) algorithm for the learning step [15]. The distributed sequential learning mechanism for a 2×2 system is proposed as follows:

1. Initialize $\theta_1(t-1)$ and $\theta_2(t-1)$;
2. RLS-1. Compute $\theta_1(t) = \theta_1(t-1) + K_1 \cdot (y_1(t) - \phi_1^T(t) \cdot \theta_1(t-1) - \phi_2^T(t) \cdot \theta_2(t-1))$ in the least squares sense and communicate to RLS-2.
3. RLS-2. Compute $\theta_2(t) = \theta_2(t-1) + K_2 \cdot (y_2(t) - \phi_2^T(t) \cdot \theta_2(t-1) - \phi_1^T(t) \cdot \theta_1(t))$ in the least squares sense and communicate to RLS-1.
4. Go to step 2 at the next sampling period.

The gain K_i can be computed recursively. Further, the error term multiplied with the gain can be weighted with an exponential forgetting factor [15]. Because the distributed RLS algorithm has the same structure as that of DN MPC, it was combined to add the *learning* feature to the DN MPC.

The DRLS algorithm itself has mild requirements such as (i) the measurement noise is assumed to be white and (ii) the parameters vary slowly and continuously, which are generally true for RLS even. In such cases, the forgetting factor is prescribed to be in between 0.98 and 1. For the distributed RLS case, if the incoming information is not uniformly distributed in the parameter space, a directional forgetting factor may be used [16]. Therefore, it is desirable to assign different forgetting factors to different parameters (however, this is to be determined based on simulation).

Remark 1

The arguments for the controller have been given for the case of two subsystems only, but same arguments apply for any finite $M > 0$ number of interconnected subsystems, where each subsystem has a copy of the plantwide model and can evaluate the objective function independently (by definition of cooperative control).

3. APPLICATION ON A HYDROSTATIC DRIVETRAIN BENCHMARK PROCESS

Generically, the hydrostatic drive uses fluid under pressure to transmit engine power in order to drive wheels or tracks. Mechanical power is converted to hydraulic power and back to mechanical power by a pump–motor synergy. The pump and motor are joined in a closed-hydraulic loop as shown in Figure 2(a), which is good for power transmission when variable output speed is required. Hydrostatic transmissions outperform electrical, gear-type transmissions as they can offer fast response, maintain precise speed under variable load, allows infinitely variable speed control from zero to infinity, and can increase torque without changing gears. In a closed-hydrostatic transmission, the torque can be transmitted in both directions, thus allowing hydrostatic breaking. However, this property implies the existence of a precise control of the traction effort and speed. Another important advantage for hydrostatic drives is the high efficiency and thereby low fuel consumption when compared with hydrodynamic drives [11].

Typically, hydraulic systems are highly nonlinear, complex dynamic plants. For this reason, linear model-based controllers that are used in practice often fail to maintain good performance. Although nonlinear differential equations can be used to describe in detail a hydraulic system, it is difficult to find suitable model-based controllers without loss of implementability [11].

3.1. Modeling

To understand the hydrostatic assembly, we would briefly describe the working principles of a motor followed by a pump-controlled motor. A hydraulic motor consists of a swash plate connected to a rotating barrel with pistons sitting on the plate via connectors. Valve plate that ports inlet fluid to half of the cylinder barrel and pistons receiving this are forced against the swash plate. This causes the barrel attached to the drive shaft to rotate. Variable displacement can be achieved by varying the angle of the swash plate. A hydraulic motor can be made to work in an opposite fashion, that is, the drive shaft rotation (due to a connected engine) now causes the pistons to generate a pressurized flow. This arrangement is called a hydraulic pump.

For an ideal hydraulic motor/pump, the mechanical power output is given by [17]

$$hp|_{out} = T_g \cdot \omega_m \quad (12)$$

where T_g is the torque generated by the motor and ω_m the angular speed of the motor shaft. The hydraulic power supplied to the motor is

$$hp|_{in} = (P_1 - P_2) \cdot Q_m \quad (13)$$

where P_1, P_2 are pressures in high, low pressure lines, respectively (refer to Figure 2(a)) and Q_m is the oil flow through the motor. Assuming 100% motor efficiency, we have that

$$T_g = S_m \cdot (P_1 - P_2) \text{ , where} \quad (14)$$

$$S_m = Q_m / \omega_m \text{ (by definition)} \quad (15)$$

where S_m is the volumetric displacement (stroke) of the motor. However, in practice, leakage flows and friction are important sources of losses. There are two types of leakage: internal and external. The internal leakage is given by

$$Q_{im} = C_{im} \cdot (P_1 - P_2) \quad (16)$$

where C_{im} is the internal leakage coefficient. The external leakage is given by

$$Q_{emi} = C_{em} \cdot P_i, \quad i \in \{1, 2\} \quad (17)$$

where C_{em} is the external leakage coefficient. Further, there are two major sources of torque losses

1. The damping torque (due to shearing the fluid). $T_d = B_m \cdot \omega_m$, where B_m is the viscous damping coefficient; and
2. The friction force opposing the motion of piston. $T_f \propto \text{Sgn}(\omega_m) \cdot (P_1 + P_2)$.

Hence, the resultant torque delivered to the load can be written as

$$T_l = S_m \cdot (P_1 - P_2) - T_d - T_f \quad (18)$$

From the continuity equation for high (modulated) pressure forward chamber, we have that (refer to Figure 2(a))

$$S_p \cdot \omega_p - C_{ip} \cdot (P_1 - P_2) - C_{ep} \cdot P_1 - C_{im} \cdot (P_1 - P_2) - C_{em} \cdot P_1 - S_m \cdot \omega_m = \frac{V_0}{\beta} \frac{dP_1}{dt} \quad (19)$$

where subscript p denotes 'pump', V_0 is the volume of forward chamber, and β is the bulk modulus of system. The torque balance equation gives

$$T_g = S_m \cdot (P_1 - P_2) = J_t \cdot \dot{\omega}_m + B_m \omega_m + T_l \quad (20)$$

where J_t is the total inertia of the motor and the load.

In the setup from Figure 2(b), we have one pump with variable displacement driving two motors with variable displacement. The pump is driven by an engine, which is speed controlled. Thus, the aforementioned analysis can be directly extended now to account for the two motors. It follows from (19) that the combined continuity equation for the high pressure line, P_1 becomes

$$\begin{aligned} S_p \cdot \omega_p - C_{ip} \cdot (P_1 - P_2) - C_{ep} \cdot P_1 - 2 \cdot C_{im} \cdot (P_1 - P_2) \\ - 2 \cdot C_{em} \cdot P_1 - S_{m1} \cdot \omega_{m1} - S_{m2} \cdot \omega_{m2} = \frac{V_0}{\beta} \frac{dP_1}{dt} \end{aligned} \quad (21)$$

Subsequently, for the low pressure line, P_2 , we have

$$\begin{aligned} S_{m1} \cdot \omega_{m1} + S_{m2} \cdot \omega_{m2} - 2 \cdot C_{im} \cdot (P_1 - P_2) - 2 \cdot C_{em} \cdot P_1 \\ - C_{ip} \cdot (P_1 - P_2) - C_{ep} \cdot P_1 - S_p \cdot \omega_p = \frac{V_0}{\beta} \frac{dP_2}{dt} \end{aligned} \quad (22)$$

The torque balance equations at two hydromotors are

$$\begin{aligned} S_{m1} \cdot (P_1 - P_2) &= J_{t1} \cdot \dot{\omega}_{m1} + B_{m1} \omega_{m1} + T_{l1} \\ S_{m2} \cdot (P_1 - P_2) &= J_{t2} \cdot \dot{\omega}_{m2} + B_{m2} \omega_{m2} + T_{l2} \end{aligned} \quad (23)$$

where subscripts 1,2 denote the first and second hydromotors, respectively. Finally, the torque balance equation for the driving electric motor is

$$T_{DrEM} = (J_{Dr} + J_p) \cdot \dot{\omega}_p + S_p \cdot (P_1 - P_2) \quad (24)$$

The Equations (21), (22), (23), and (24) define the model of the hydrostatic drivetrain.

3.2. Open-loop tests

In the drivetrain setup from Figure 2(b), the pump speed driven by the engine and the displacement volume of the pump are fixed. Thus, the control objective is to achieve the desired hydromotor speed setpoints by manipulating the associated displacement volumes of the two motors while minimizing the effect of the unmeasured load torque disturbances. In order to access the characteristics of this system, we use the aforementioned derived system equations together with the parameters obtained from the real machine to demonstrate the presence of coupled and nonlinear dynamics.

In the first test, the displacement volume of the second hydromotor is fixed and that of the first hydromotor is varied stepwise. As shown in Figure 3(a), changing the displacement volume of the first hydromotor is almost equal but opposite influence on the speeds of both hydromotors. This observation suggests that the hydrostat is composed of highly coupled subsystems.

In the second test, both hydromotor displacement volumes are changed stepwise. Because both hydromotors are assumed to have the same physical behavior, the effects of nonlinearity will be more pronounced if the directions of the simultaneous variation in the inputs are the same. As depicted in Figure 3(b), the step changes in the hydromotor displacement volumes produce significantly different dynamics in the hydromotor speeds. These differences are in terms of varying gain, damping coefficients, and time constants, all depending on the operating point. This observation suggests the presence of significant nonlinearities in the global system.

3.3. Closed-loop tests

In order to explicitly demonstrate the superiority of Algorithm 1 in terms of performance and computation, a centralized NEPSAC controller is designed with $N_1 = 1, N_u = 2, N_2 = 5, \Lambda = 10 \cdot I$. Our target is now to show that two distributed NEPSAC controllers with the same parameters $N_1 = 1, N_u = 1, N_2 = 5, \Lambda = 10$ approach the centralized performance, which is the best that can be achieved as all information is available. The DN MPC controllers communicate sequentially and only once as in Algorithm 1. The control penalty is adapted to a very high value whenever the cost does not decrease monotonically, essentially freezing the new control to the past value. The sampling time is 100 ms, and the inputs are subjected to actuator constraints of 12–100% limits on strokes. The goal is to drive the hydromotor speeds to the respective setpoints. The results are plotted in Figure 4(a). As can be inferred, there is no difference to the naked eye between the performance of the CN MPC and DN MPC tracking controllers with the controls almost overlapping each other. This is a validation of Theorem 1 and Lemma 1 as the distributed locally optimal solution approaches the centralized one. The graphical verification is given in Figure 4(b), where it can be seen that the first steps are enough to guarantee sufficient decrease in the gradient, which is in fact the case here.

Next, the computational costs for both the methods are compared. The maximum time, average time required to perform all the computations within one sampling time are 500 ms and 20 ms for CN MPC and 50ms and 2ms for DN MPC on embedded MATLAB for real-time target. This clearly shows that as the peak computation time for CN MPC of 500 ms is greater than the sampling time of 100 ms, it cannot be used in practice. However, the DN MPC controller is well within the limits even in the worst case.

4. EXPERIMENTAL RESULTS

The experimental setup as shown in Figure 2(b) consists of (i) a speed controlled driving motor; (ii) two torque controlled load motors; and (iii) a hydraulic pump attached to the engine which is connected to the two hydromotors via flywheels. The stroke of the pump is fixed to 40% and the speed of the engine to 1200 rpm. The objective is to regulate the hydromotor speeds, reject load disturbances, and adapt to effects of varying load. Input constraints are set on the actuators, that is, the stroke of the two hydromotors must be between 12% and 100%. First, we present the design of two PID loops, which would be used for comparison purposes.

4.1. Quasi-decoupled PID control

It is clear from the analysis in Section 3.2 that there exists no unique relation between the stroke of one hydraulic motor and its speed. Therefore, it is impossible to determine steady-state gains necessary for classic PID tuning methods. A transformation is thus required that decouples significantly the hydromotor interaction and is discussed hereafter.

One can rewrite the flow equation (assuming incompressible fluid and negligible leakage) as

$$S_p \cdot \omega_p = (S_{m1} + S_{m2}) \cdot (\omega_{m1} + \omega_{m2}) + (S_{m1} - S_{m2}) \cdot (\omega_{m1} - \omega_{m2}) \quad (25)$$

Ignoring the second term yields

$$d(\omega_{m1} + \omega_{m2})/d(S_{m1} + S_{m2}) < 0 \quad (26)$$

Because the pressures across the pump and the two motors are equal, we have

$$d(S_{m1}/S_{m2}) = d(T_{g1}/T_{g2}) = (T_{g2} \cdot dT_{g1} - T_{g1} \cdot dT_{g2})/T_{g2}^2 \quad (27)$$

where T_{g1}, T_{g2} are the torques of the two hydromotors. Assuming positive torques in steady state, if S_{m1}/S_{m2} increases, then either T_{g1} increases or T_{g2} decreases. Further, at the flywheels, $dT_{l1}/d\omega_{m1} > 0$ and $dT_{l2}/d\omega_{m2} > 0$. This leads to the resulting steady-state relation

$$d(\omega_{m1}/\omega_{m2})/d(S_{m1}/S_{m2}) > 0 \quad (28)$$

The relations (26) and (28) concur that two independent PID controllers can be designed with one controlling the sum of hydromotor speeds with the sum of hydromotor strokes and the other controlling the ratio of hydromotor speeds with ratio of hydromotor strokes. The PID controllers are defined on the sum and the ratio error signals. The controller outputs are then re-transformed to the hydromotor strokes through negative exponentials (as it more or less neutralizes the nonlinear stroke–speed relation). This is indeed by far the most effective PID design, as the others which fail to induce any decoupling and makes the system unstable and not safe for real tests.

4.2. Experimental validation

Two identical DNMPC controllers have been designed with prediction horizon $N_1 = 1$ to $N_2 = 5$, control horizon $N_u = 1$, and control penalty $\Lambda = 10$, an integrator as disturbance filter, that is, $C/D = 1/(1 - q^{-1})$. The embedded DRLS uses a forgetting factor of 0.99 and the closed loop is sampled at $T_s = 100$ ms. The DNMPC uses a cooperative cost function and exchanges the optimal trajectories once every sampling period. The number of iterations in the nonlinear EPSAC algorithm within the DNMPCs have been restricted to 1 iteration per sampling period.

First, the proposed control strategy has been tested on the real setup with nominal settings, that is, temperature of 50°C and load torques at 20 Nm. During this first closed-loop test, variations in the speed setpoints are tested, and as a disturbance, the load torque values are varied within ± 10 Nm.

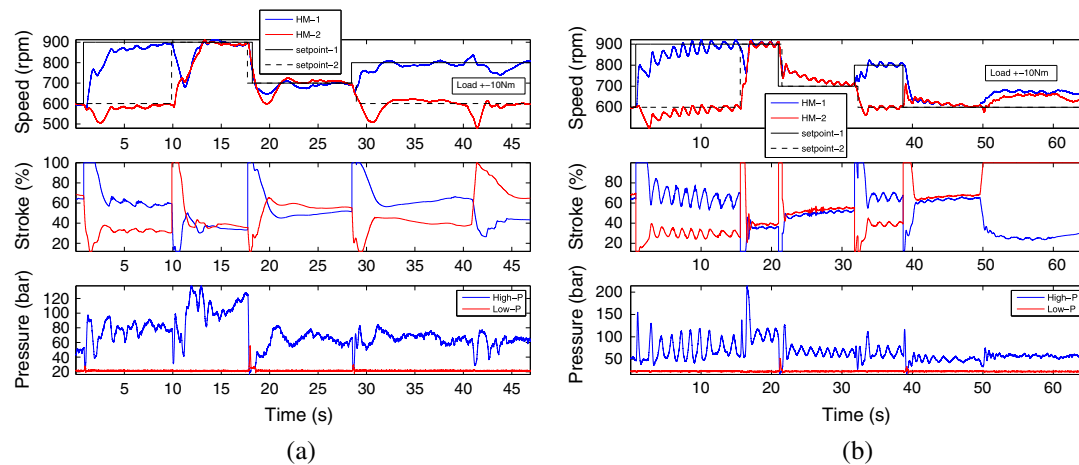


Figure 5. (a) Tracking performance and disturbance rejection by distributed NMPC and (b) PID control.

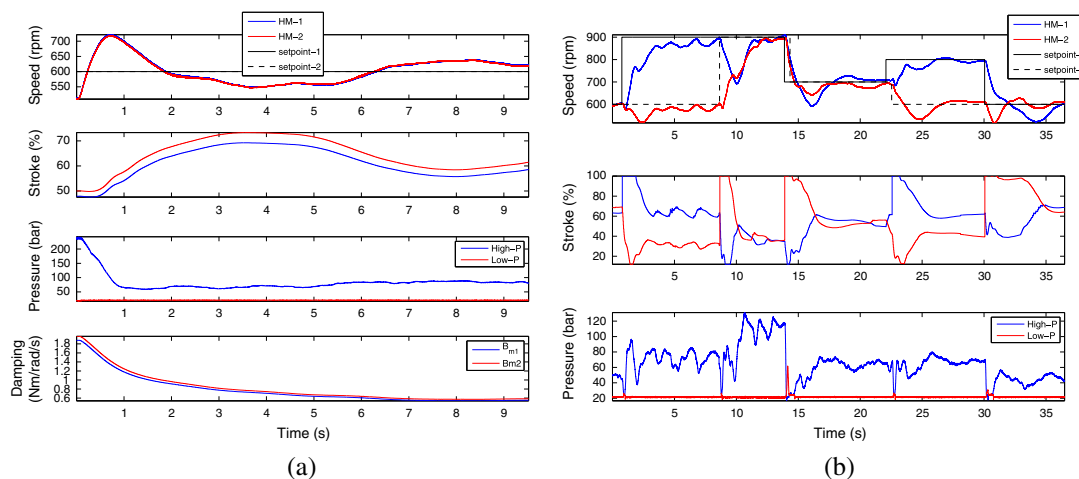


Figure 6. (a) Learning the true motor damping coefficients by the distributed RLS method and (b) robust performance of the distributed NMPC after learning the correct damping coefficients.

The performance of the DNMPC is given in Figure 5(a). For comparison purpose, the same experiment has been evaluated with the PID controllers and the results given in Figure 5(b). It can be observed that the DNMPC outperforms the PID control, generating a much smoother control action compared with the rather oscillatory response of the PID control strategy. The net oil flow, which is the product of stroke and angular velocity, must remain constant. This implies that if both speeds go up, the strokes go down (similarly for reverse sign). The load change is nicely compensated by the DNMPC, whereas significant offsets are present in case of PID control. Both in nominal and perturbed load settings, the DNMPC controller manages a settling time < 10 s with a rise time < 5 s, whereas the PID though has the same rise time of < 5 s; the settling time $\rightarrow \infty$ because of sustained oscillations in the nominal case and the PID produce a big steady-state error in the perturbed case, that is, after 50 s.

In the next experiment, the controllers have been initialized with erroneous motor damping coefficients, that is, 1.9 Nm/rad/s. Given that damping varies exponentially with temperature, the ability to continuously learn the right damping is of high practical importance. As shown in Figure 6(a), the distributed RLS algorithm converges to the true value of 0.55 Nm/rad/s in a reasonable period. Consequently, the performance of the controller in terms of settling time and rise time (plotted in Figure 6(b)) remains comparable with the nominal case, demonstrating robustness of the proposed control strategy.

5. CONCLUSIONS

A novel DNMPC framework has been presented together with an RLS distributed learning method. The main contribution ensured guaranteed improvement in the cost function by the DNMPC even in the limiting non-iterative case and a tenfold reduction in computation time for nonlinear non-convex problems. The first principle modeling of the hydrostatic drivetrain has been developed and used for the design of distributed control of the coupled nonlinear benchmark system. Experimental validation on the benchmark suggests that the proposed control methodology is successful in practice.

ACKNOWLEDGEMENT

Abhishek Dutta is supported within the framework of the LeCoPro project (grant no. 80032) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). We are grateful to the technical support provided by Flanders Mechatronics Research Center during the experiments conducted on the hydrostat setup.

REFERENCES

1. Scattolini R. Architectures for distributed and hierarchical model predictive control—a review. *Journal of Process Control* 2009; **19**(5):723–731.
2. Christofides PD, Liu J, de la Peana DM. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Springer-Verlag: London, 2011.
3. Picasso B, Romani C, Scattolini R. Tracking control of wiener models with hierarchical and switching model predictive control. *Optimal Control Applications and Methods* 2013; **34**(1):1–16.
4. Liu J, Munoz de la Pena D, Christofides PD. Distributed model predictive control of nonlinear process systems. *AIChE Journal* 2009; **55**(5):1171–1184.
5. Richards A, How JP. Robust distributed model predictive control. *International Journal of Control* 2007; **80**(9):1517–1531.
6. Camponogara E, Jia D, Talukdar BHKS. Distributed model predictive control. *IEEE Control Systems* 2002; **22**(1):44–52.
7. Camacho EF, Bordons C. *Model Predictive Control*. Springer-Verlag: London, 2003.
8. Rawlings JB, Stewart BT. Coordinating multiple optimization-based controllers: new opportunities and challenges. *Journal of Process Control* 2008; **18**(9):839–845.
9. Christofides PD, Scattolini R, de la Pena DM, Liu J. Distributed model predictive control: a tutorial review and future research directions. *Computers & Chemical Engineering* 2013; **51**:21–41.
10. De Keyser R. *Model Based Predictive Control for Linear Systems*, UNESCO Encyclopaedia of Life Support Systems. Article contribution 6.43.16.1. Eolss Publishers Co Ltd: Oxford, 2003; 35.

11. Rydberg K. Hydrostatic drives in heavy mobile machinery-new concepts and development trends. *SAE Transactions* 1998; **107**:232–238.
12. Maestre JM, Munoz de la Pena D, Camacho EF. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods* 2011; **32**(2):153–176.
13. Venkat AN, Rawlings JB, Wright SJ. Stability and optimality of distributed model predictive control. *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05*: IEEE, Seville, 2005; 6680–6685.
14. De Keyser RMC, Van Cauwenberghe AR. A self-tuning multistep predictor application. *Automatica* 1981; **17**(1):167–174.
15. Ljung L. *System Identification: Theory for the User* (2nd edn). Prentice Hall PTR: New Jersey, 1998.
16. Bittanti S, Bolzern P, Campi M. Convergence and exponential convergence of identification algorithms with directional forgetting factor. *Automatica* 1990; **26**(5):929–932.
17. Merritt HE. *Hydraulic Control Systems*. Wiley: New York, 1967.